

U.S.P.S. Express Mail Label No.: EV 164034194 US

Date of Deposit: January 14, 2004

Attorney Docket No. 15354US01

IMPROVED DATA RETRIEVAL FROM A STORAGE DEVICE USING A COMBINED  
ERROR CORRECTION AND DETECTION APPROACH

RELATED APPLICATIONS / INCORPORATION BY REFERENCE

[01] [Not Applicable]

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[02] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[03] [Not Applicable]

## BACKGROUND OF THE INVENTION

[04] The basis of error detection and error correction of data stored in storage devices relies in the incorporation of redundant information to the data. The redundant information (also known as parity or redundancy symbols) is used by special hardware and / or software to detect and correct errors in order to recover missing or corrupt data within a media of the storage device. The data residing in a storage device, such as a magnetic disk drive, is encoded using two types of codes: an error correction code (ECC) and an error detection code (EDC). ECCs are used to correct errors while EDCs are used to detect errors in data as it is read from the storage device. Unfortunately, in many instances, such ECCs and EDCs utilize algorithms, implemented either in hardware or firmware, that are incapable of recovering the data in the storage device.

[05] When employing ECC and EDC to data stored in a magnetic disk drive, the number of errors (e.g., symbols errors) capable of being corrected in a sector of data is limited to a specific maximum number. Unfortunately, when the number of such errors exceeds this maximum number, the data associated with

this sector may not be readable. As a consequence, critical data may never be recovered from the magnetic disk drive.

[06] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of skill in the art, through comparison of such systems with some aspects of the present invention as set forth in the remainder of the present application with reference to the drawings.

## BRIEF SUMMARY OF THE INVENTION

[07] Aspects of the invention provide for a more effective method and system of reading data stored in a media of a storage device by way of correcting and detecting errors using a three stage decoding process.

[08] In one embodiment, a method of encoding data stored in a storage device comprises generating a first polynomial whose roots comprise one or more powers of a primitive element of a Galois field, generating a second polynomial whose roots comprise one or more powers of a primitive element of the Galois field and generating a product of the first polynomial and the second polynomial.

[09] In one embodiment, a method of decoding data stored in a storage device comprises reading an encoded codeword from the storage device, performing a first division of the encoded codeword by a first polynomial, determining whether a first remainder exists from the first division, first correcting the encoded codeword using the first remainder, performing a second division of the encoded codeword by a second polynomial, determining

whether a second remainder exists from the second division, performing a third division of the encoded codeword by a product of the first and the second polynomials if the second remainder exists from the second division, determining a third remainder from the third division, and second correcting the encoded codeword using the third remainder.

[10] In one embodiment, a method of error correction and detection of data stored in a storage device comprises generating a first polynomial whose roots comprise one or more consecutive powers of a primitive element of a Galois field, generating a second polynomial whose roots comprise one or more consecutive powers of a primitive element of the Galois field, and generating a product of the first polynomial and the second polynomial. Subsequently, a first division is performed on a third polynomial by the product to generate a first remainder wherein the first remainder is used to generate an encoded codeword, wherein the third polynomial is used to represent a sector of data, in which the sector of data comprises a number of symbols. Subsequently the encoded codeword is written into a media of the storage device. The method of error detection and correction further comprises reading the encoded codeword

from the storage device, performing a second division of the encoded codeword by the first polynomial, determining whether a second remainder exists from the second division, correcting the encoded codeword using the second remainder, performing a third division of the encoded codeword by the second polynomial, determining whether a third remainder exists from the third division, performing a fourth division of the encoded codeword by a product of the first and the second polynomials if the third remainder exists from the second division, determining a fourth remainder from the fourth division, and correcting the encoded codeword using the fourth remainder.

[11] In one embodiment, a system to effectively correct and detect errors in a media of a storage device comprises an encoder for encoding data that is written onto the media of the storage device, a decoder for decoding the data that is read from the media of the storage device using a process that utilizes at least two stages of error correction.

[12] These and other advantages, aspects, and novel features of the present invention, as well as details of illustrated embodiments, thereof, will be more fully understood from the following description and drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

[13] Figure 1 is a block diagram of an error correction and detection system in accordance with an embodiment of the invention.

[14] Figure 2 is a flow diagram of an encoding process used to generate an encoded codeword in accordance with an embodiment of the invention.

[15] Figures 3A and 3B are flow diagrams of a decoding process used to decode an encoded codeword in accordance with an embodiment of the invention.



## DETAILED DESCRIPTION OF THE INVENTION

[16] Aspects of the invention provide for a more effective method and system of retrieving data stored in a media of a storage device by way of correcting and detecting errors using a multi-stage decoding process. In one embodiment, the storage device comprises a magnetic hard drive. The data errors are corrected and detected when reading the data from the storage device. In one embodiment of the present invention, error correction and detection is performed by way of the Reed-Solomon code. In one embodiment, the system and method applies an encoding/decoding technique that allows error correction and detection to be performed over a number of successive decode stages or processing stages. In one embodiment, three successive (decode) processing stages are implemented. In one embodiment, at least two error correction processing stages are implemented. Use of the system and method increases the maximum number of symbol errors that may be corrected in an encoded codeword, providing an improvement in data recovery.

[17] Figure 1 is a block diagram of an error correction and detection system in accordance with an embodiment of the invention. The system comprises an encoder/decoder subsystem 104, a storage device 116, a memory 124, a control processor (CPU) 128, and a data interface 132. The encoder/decoder subsystem 104 comprises an encoder/decoder circuitry 108 and an encoder/decoder memory 112. The encoder/decoder circuitry 108 may comprise digital logic circuitry capable of encoding data it receives from the data interface 132. In addition, the encoder/decoder memory 112 may store a set of instructions, software, or firmware capable of performing encoding and decoding of data it receives from one or more devices (e.g., the data interface 132, storage device 116, memory 124, CPU 128). The encoding and decoding of data performed by encoder / decoder circuitry 108 may be facilitated by executing one or more instructions resident within the encoder/decoder memory 112. In one embodiment, the encoder/decoder memory 112 may store a set of instructions resident in the encoder/decoder memory 112 capable of encoding and decoding data using one or more algorithms. The one or more algorithms may be implemented when the CPU executes the set of instructions.

The one or more algorithms implemented may comprise, for example, the encoding and/or decoding algorithms that will be described in Figures 2, 3A, and 3B. The encoder/decoder memory 112 may comprise a read only memory (ROM) such as a programmable read only memory (PROM). In one embodiment, the encoder/decoder circuitry 108 and/or the encoder/decoder memory 112 may implement a Reed–Solomon encoder/decoder. For example, the encoder/decoder circuitry 108 and/or the encoder/decoder memory 112 may generate a number of encoded codewords, such as Reed–Solomon codewords, after receiving data from the data interface 132. When the encoder/decoder subsystem 104 writes to the storage device 116, one or more appropriate write control signals are transmitted to the storage device 116. On the other hand, when the encoder/decoder subsystem 104 reads from the storage device 116, one or more appropriate control signals are transmitted to the storage device 116. In both a write or read operation, data is transmitted between the encoder/decoder subsystem 104 and the storage device 116. The storage device 116 writes into or reads from a media 120. The media 120 resides within the storage device 116 and may comprise a removable or portable media

120. The media 120 may comprise a magnetic media such as that found in a magnetic disk drive. The media 120 stores one or more encoded codewords generated by the encoder/decoder subsystem 104. The memory 124 may be used by the encoder/decoder subsystem 104 and the control processor 128 for storing and manipulating data during the encoding and decoding process. It is contemplated the memory 124 may comprise random access memory (RAM) such as dynamic RAM (DRAM).

[18] Each of the encoded codewords generated by the encoder/decoder subsystem 104 comprises a number of symbols. Each symbol comprises any number of bits. In one embodiment, the number of bits in a symbol equals 10 while the maximum number of symbols in an encoded codeword equals 1023. The data received by the encoder/decoder subsystem is partitioned into one or more sectors of data. Each sector of data may be represented by an encoded codeword. The encoded codeword comprises one or more data symbols and one or more redundancy or parity symbols. The redundancy or parity symbols may be appended after the one or more data symbols. The redundancy or

parity symbols may comprise symbols generated by an exemplary Reed-Solomon encoder.

[19] In one embodiment, if  $n$  denotes the number of data and parity symbols in an encoded codeword (required to represent the sector of data), the number of parity symbols used in the encoded codeword is equal to  $2(t+k)$ . In one embodiment, the error correction and detection system generates a first polynomial of degree  $2t$  and a second polynomial of degree  $2k$ . The roots of the first polynomial as well as the roots of the second polynomial comprise consecutive powers of a primitive element of a Galois field. In one embodiment, the first root of the second polynomial has power that is consecutive to that of the last root of the first polynomial. Mathematically, the previously described relationships may be represented by the following equations:

$$[20] \quad p(x) = (x - \alpha^1) \dots (x - \alpha^{2k}) \quad [\text{first polynomial}]$$

$$[21] \quad f(x) = (x - \alpha^{2k+1}) \dots (x - \alpha^{2(t+k)}) \quad [\text{second polynomial}]$$

[22] As described by the previous equations, the primitive element of the Galois field is represented by  $\alpha$ . The encoder/decoder subsystem 104 utilizes a third polynomial, termed the generator polynomial, which corresponds to the product of the first polynomial,  $p(x)$ , and the second polynomial,  $f(x)$ . The generator polynomial,  $g(x)$ , may be represented by the following equation:

$$[23] \quad g(x) = p(x)f(x) = (x - \alpha^1) \dots (x - \alpha^{2(t+k)}) \quad [\text{third polynomial}]$$

[24] The one or more data symbols,  $c_{2(t+k)}$ ,  $c_{2(t+k)+1}$ , ...,  $c_{n-1}$  of the encoded codeword may be represented by the following equation:

$$[25] \quad C(x) = c_{2(t+k)}x^{2(t+k)} + c_{2(t+k)+1}x^{2(t+k)+1} + \dots + c_{n-1}x^{n-1}$$

[26] After  $C(x)$  is divided by  $g(x)$ , the remainder that results provides the one or more parity symbols of the encoded codeword. The remainder polynomial,  $r(x)$ , of degree  $2(t+k)-1$ , may be represented by the following equation:

$$[27] \quad r(x) = r_0 + r_1x + \dots + r_{2(t+k)-1}x^{2(t+k)-1}$$

[28] The encoded codeword (e.g., the one or more data and parity symbols) is generated by appending  $r(x)$  to  $C(x)$  and may be represented by the following equation.

$$[29] \quad C^*(x) = C(x) + r(x) = r_0 + r_1x + \dots + r_{2(t+k)-1}x^{2(t+k)-1} + c_{2(t+k)}x^{2(t+k)} + c_{2(t+k)+1}x^{2(t+k)+1}$$

$$[30] \quad + \dots + c_{n-1}x^{n-1}$$

As shown, the coefficients of  $C^*(x)$  represent a total of  $n$  symbols.

[31] Figure 2 is a flow diagram of an encoding process or encoding algorithm used to generate an encoded codeword in accordance with an embodiment of the invention. The encoding process or algorithm utilizes the polynomial equations,  $f(x)$ ,  $p(x)$ ,  $g(x)$ ,  $r(x)$ ,  $C(x)$ , and  $C^*(x)$  previously described, each of which are defined over a Galois field. The encoding algorithm utilizes a process such that decoding may be performed sequentially over multiple processing stages, as will be described later in Figures 3A and 3B, for example. The encoding process starts at step 204, in which,  $f(x)$  is determined based on  $t$ . The encoder/decoder subsystem, as referenced in Figure 1, generates  $f(x)$  as a polynomial of degree  $2t$  having roots which are consecutive powers of a primitive element in a Galois field. The variable  $t$  defines the maximum number of symbol errors capable of being corrected by the error correction and detection system shown in Figure 1 when the polynomial  $f(x)$  is used. In one

embodiment, the variable  $t+k$  is defined as one-half the difference between the total number of symbols in a codeword and the number of data symbols in the codeword. As described earlier,  $f(x)$  comprises a polynomial of degree  $2t$ . In one embodiment,  $t=24$  or  $2t=48$ . At step 208, the encoder/decoder subsystem generates  $p(x)$  as a polynomial of degree  $2k$ , in which its roots are consecutive powers of the same primitive element. In one embodiment,  $k=2$  or  $2k=4$ . The variable  $k$  defines the maximum number of symbol errors capable of being corrected by the error correction and detection system shown in Figure 1 when the polynomial  $p(x)$  is used. In one embodiment, the polynomials  $f(x)$  and  $p(x)$  are pre-computed and implemented at the time of manufacture (e.g., at the time the error correction and detection system, as described in Figure 1, is built or manufactured). For example,  $\alpha$  and the orders of  $f(x)$  and  $p(x)$  are determined only once, at the time the error correction and detection system is built. At step 212, the encoder/decoder subsystem generates the generator polynomial,  $g(x)$ . As described earlier,  $g(x)$  is the product of  $f(x)$  and  $p(x)$ . The polynomials,  $f(x)$ ,  $p(x)$ , and  $g(x)$  may be defined and generated by way of a Reed-Solomon code encoder. In one embodiment, the polynomial  $g(x)$  is



predetermined since, as described previously,  $f(x)$  and  $p(x)$  are pre-computed at the time of manufacture. At step 216, the remainder polynomial,  $r(x)$ , is generated by dividing  $C(x)$  using  $g(x)$ . The remainder polynomial,  $r(x)$ , contains  $2(t+k)$  parity or redundancy terms. Then at step 220, the encoded codeword,  $C^*(x)$  is generated by appending the  $2(t+k)$  parity or redundancy symbols to the data portion of the codeword,  $C(x)$ .  $C(x)$  is a polynomial defined over a Galois field whose coefficients represent one or more data symbols. Thereafter, the encoded codeword,  $C^*(x)$ , may be written into a media of a storage device such as an exemplary magnetic disk drive.  $C^*(x)$  is a polynomial defined over a Galois field whose coefficients represent one or more data and parity (or redundancy) symbols. Aspects of the invention provide for the computation of  $C^*(x)$ , for each  $C(x)$  received by the error correction and detection system, using the pre-computed values for  $f(x)$  and  $p(x)$ . In one operational embodiment, each of the one or more symbols represented by  $C(x)$  and  $C^*(x)$  comprise 10 bits.

[32] Figures 3A and 3B are flow diagrams of a decoding process or decoding algorithm used to decode an encoded codeword in accordance with an

embodiment of the invention. The decoding process or algorithm utilizes the polynomial equations,  $f(x)$ ,  $p(x)$ ,  $g(x)$ ,  $r(x)$ ,  $C(x)$ , and  $C^*(x)$  previously described. The decoding algorithm utilizes at least two error correction stages in order to more effectively correct symbol errors when reading data from media of a storage device. In one embodiment, the decoding algorithm utilizes a 3 stage decoding process in which 1) a first error correction is performed at the first processing stage using  $f(x)$ , 2) an error detection check is performed at the second processing stage using  $p(x)$ , and 3) a second error correction is performed at the third processing stage using  $g(x)$ . The first processing stage occurs at step 304, in which the encoded codeword,  $C^*(x)$ , is divided by the polynomial  $f(x)$  described in relation to Figure 2. At step 308, the decoder circuitry determines if a remainder exists after  $C^*(x)$  is divided by  $f(x)$ . If a remainder exists, the process proceeds to step 312, in which up to  $t$  symbol errors are corrected. Otherwise, if no remainder exists, the process proceeds to step 316, in order to perform an error check. During the first processing stage, the encoded codeword,  $C^*(x)$ , is corrected for up to a maximum of  $t$  errors, by way of utilizing the remainder. In one embodiment, a Reed–Solomon

algorithm is used to correct up to  $t$  errors in the encoded codeword,  $C^*(x)$ . ( $C^*(x)$  is divisible by  $g(x)$ , so it is also divisible by  $f(x)$ , in which  $f(x)$  has degree  $2t$ ; as a result, the first processing stage utilizes this fact to correct up to  $t$  symbol errors). At step 314, the decoder determines whether the encoded codeword has been successfully decoded. If the encoded codeword has not been successfully decoded, the process proceeds with step 324. Otherwise, the process proceeds with step 316, in which an error detection check is performed. The second processing stage occurs at step 316, in which the encoded codeword is divided by polynomial  $p(x)$ , in order to perform an error detection check. The one or more symbol error(s) that are corrected at step 312 may be verified by way of the error detection check. If, at step 320, it is determined that a remainder exists after division by  $p(x)$ , one or more errors still remain present in the encoded codeword  $C^*(x)$ , and the process proceeds with step 324. The second processing stage determines whether any errors exist after processing is performed at the first processing stage, as described above. However, if, at step 320, there is no remainder after division by  $p(x)$ , all errors in  $C^*(x)$  have been corrected and the process ends with a successful

decode. In one embodiment, the error detection provided by the second processing stage is implemented using a Reed–Solomon algorithm employing  $2k$  parity or redundancy symbols. If a remainder exists after division by  $p(x)$ , the process proceeds with the third processing stage at step 324, in which, the encoded codeword,  $C^*(x)$ , is divided by the generator polynomial,  $g(x)$ , as described in reference to Figure 2. As described previously, the generator polynomial  $g(x)$  has degree equal to  $2(t+k)$ . At step 324, it is determined whether a remainder exists after dividing by  $g(x)$ . If a remainder exists, at step 328, the remainder is used to correct up to  $(t+k)$  symbol errors in  $C^*(x)$ . In one embodiment, a Reed–Solomon algorithm is used to correct up to  $(t+k)$  errors in  $C^*(x)$ .  $C^*(x)$  is divisible by  $g(x)$ , in which  $g(x)$  has degree  $2(t+k)$ . As a consequence, the third processing stage utilizes this fact to correct up to  $(t+k)$  symbol errors. The error correction capability provided when dividing by  $g(x)$  allows error correction for up to a maximum of  $(t+k)$  errors. At step 332, it is determined whether a successful decode may be accomplished. If the data cannot be decoded, an unsuccessful decode or failure occurs and the process ends. Otherwise, a successful decode results, and the process ends. It is

contemplated that the error correction and detection process previously described may be implemented using a number of stages beyond the 3 stage process described, in order to provide additional error correction / detection capabilities.

[33] Alternatively, in another embodiment of the above described encoding /decoding process, a modification of a write or a read operation is performed in order to obviate an issue in which a read operation encounters a displaced sync mark an integer number of symbols before or after an actual sync mark location. In one embodiment, a cyclic shift in an encoded codeword, resulting from such a displaced sync mark, may result in another valid codeword, for example, when a Reed–Solomon codeword is used. A symbol (or byte, when symbol comprises 8 bits) number is added to the corresponding symbol (or byte) value. For example, the following mapping is performed after the previously described encoding, but prior to writing the encoded codeword onto a media of a storage device:

[34]  $c^* \rightarrow c^*_j + j, j = 0, \dots, n-1$

Here,  $c^*$  represents the symbol while  $j$  represents the symbol number. In this embodiment, when reading data from a media of a storage device, the mapping is performed prior to performing a decode of the encoded codeword, in order to correctly read the data.

[35] While the invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the invention without departing from its scope. Therefore, it is intended that the invention not be limited to the particular embodiment disclosed, but that the invention will include all embodiments falling within the scope of the appended claims.